

Create Tables:

Issued SQL commands to create new tables in the sqlite database

[2018-04-03 21:35:56] Connected

```
sql> create table R1
```

```
(  
  A INTEGER,  
  B INTEGER )
```

[2018-04-03 21:35:56] completed in 9ms

```
sql> create table R2
```

```
(  
  A INTEGER,  
  B INTEGER )
```

[2018-04-03 21:35:56] completed in 7ms

```
sql> create table R3
```

```
(  
  A INTEGER,  
  B INTEGER )
```

[2018-04-03 21:35:56] completed in 9ms

```
sql> create table R4
```

```
(  
  A INTEGER,  
  B INTEGER )
```

[2018-04-03 21:35:56] completed in 7ms

```
sql> create table R5
```

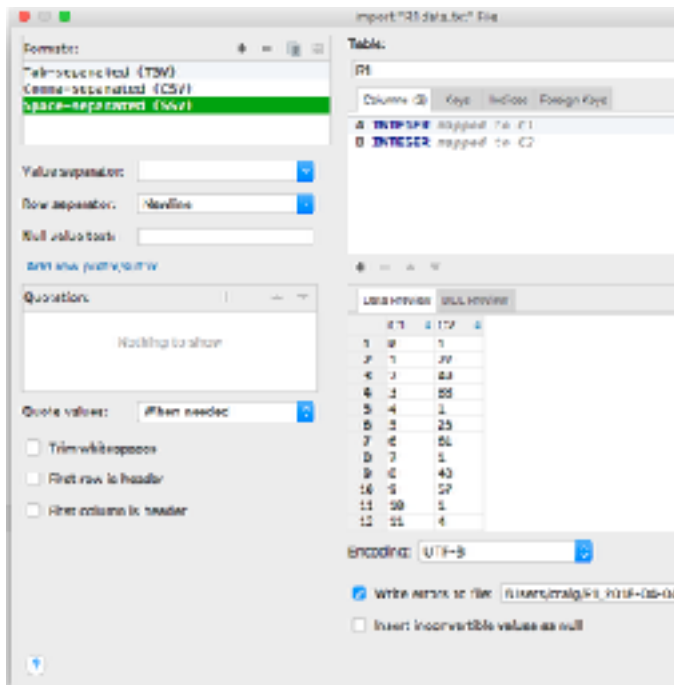
```
(  
  A INTEGER,  
  B INTEGER )
```

[2018-04-03 21:35:57] completed in 5ms

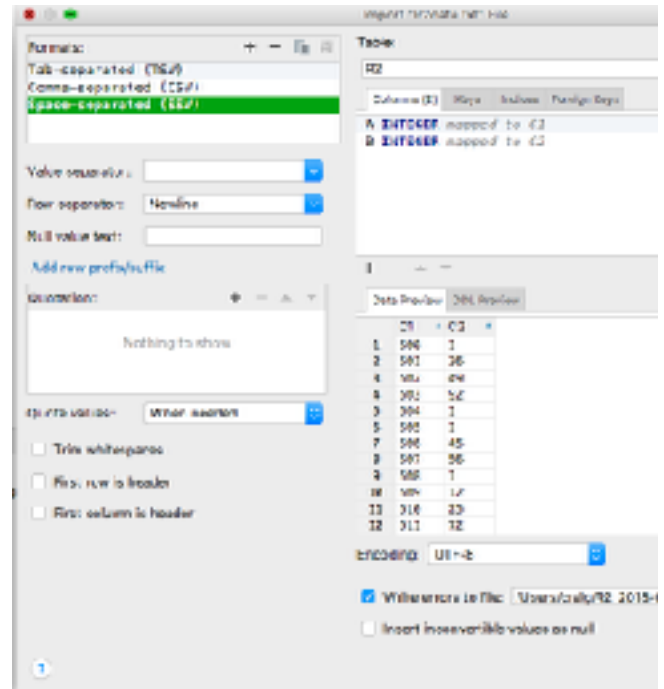
Load Operation: I had a lot of problems getting sqlite to load files, so I used the IntelliJ IDE to load files into the database. Below is the way I did, and the success of each action.

```
sqlite> .separator ';'
sqlite> .import R1data.txt R1
sqlite> .read R1
```

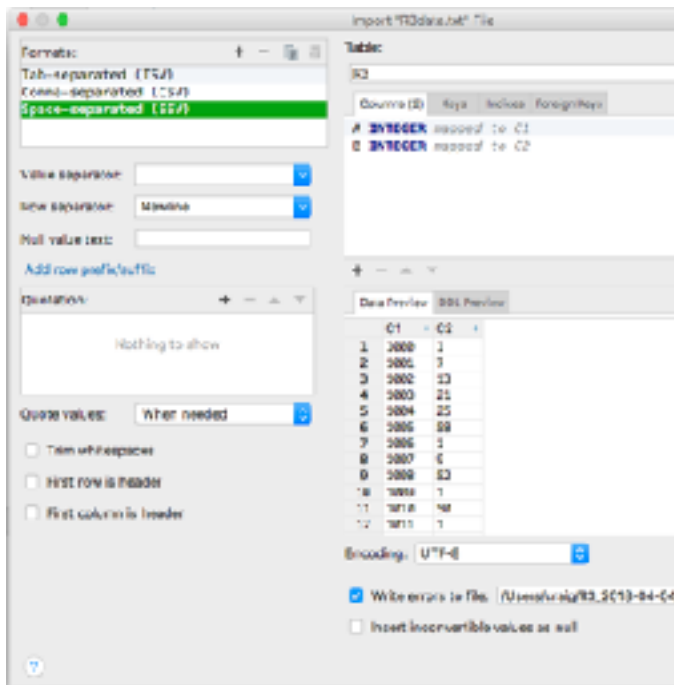
My poor attempt to use sqlite from the command line to load files to a table



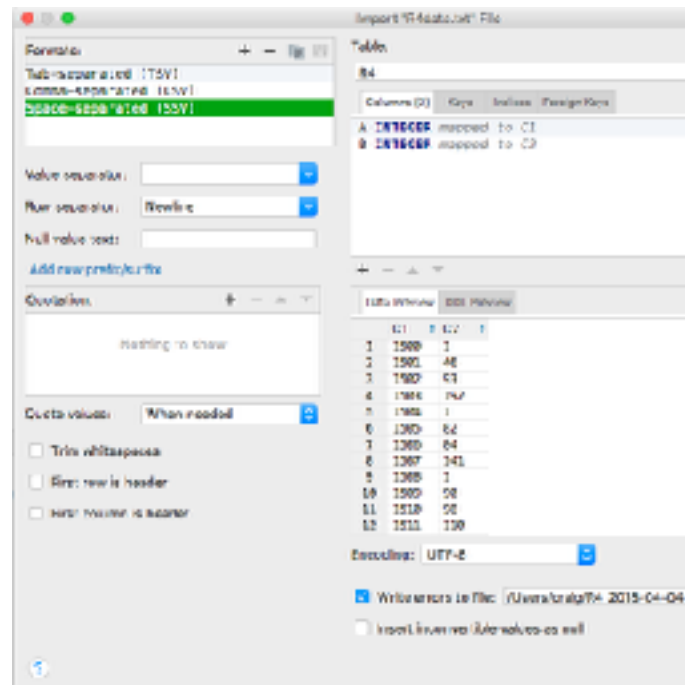
Menu to select how the data is added for R1



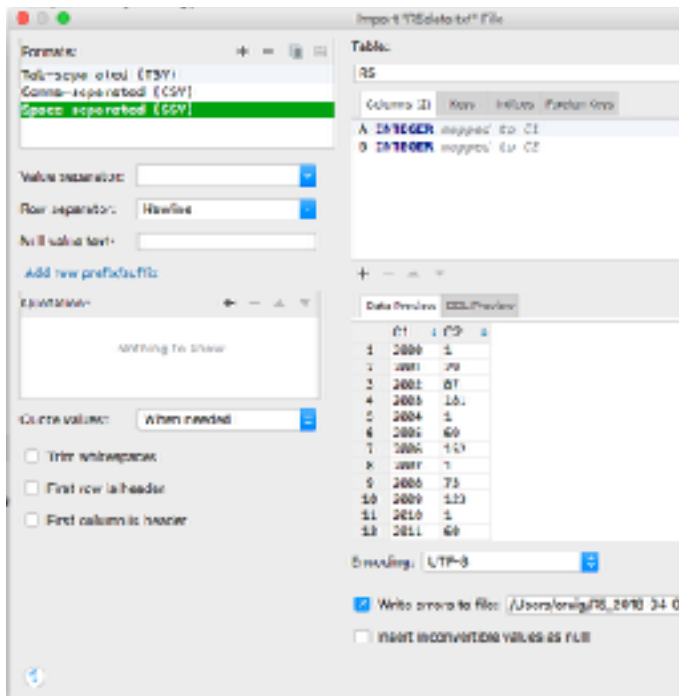
Menu to select how the data is added for R2



Menu to select how the data is added for R3



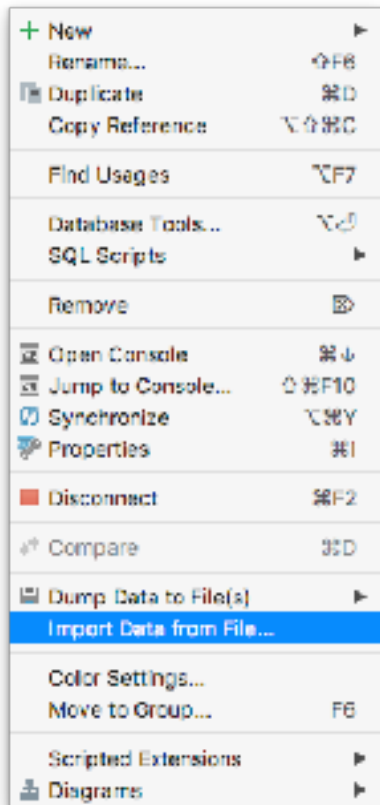
Menu to select how the data is added for R4



Menu to select how the data is added for R5



Success of importing from the files



Menu from IntelliJ. This selection is done for each table in the Database

Explain Plan: SQL commands to explain how the queries are done. Unfortunately, I couldn't get much information out of the commands that I ran in the IDE. Again, I had too many problems doing this in the command line.

--Plan1

EXPLAIN QUERY PLAN

```
select *
from R1, R2, R3, R4, R5
where
  R1.B=R2.B and R2.B=R3.B and R3.B=R4.B and R4.B=R5.B and
  R5.B=51;
```

--Plan2

EXPLAIN QUERY PLAN

```
select *
from R1, R2, R3, R4, R5
where R5.B=51 and
  R1.B=R2.B and R2.B=R3.B and
  R3.B=R4.B and R4.B=R5.B;
```

--Plan3

EXPLAIN QUERY PLAN

```
select *
from R1, R1, R2, R3, R5
where
  R5.B=51 and R1.B=R2.B and R2.B=R3.B and
  R3.B=R4.B and R4.B=R5.B;
```

--Plan4

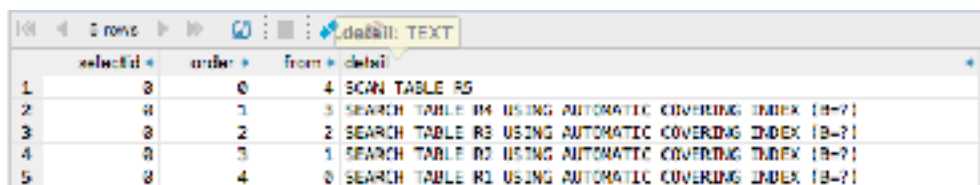
EXPLAIN QUERY PLAN

```
select *
from R1,R2, R3,R4, R5
where
  R1.B=R2.B and R2.B=R3.B and
  R3.B=R4.B and R4.B=R5.B and not ( R5.B < 51 or R5.B > 51);
```

--Plan5

EXPLAIN QUERY PLAN

```
select *
from R1, R2, R3, R4, R5
where
  R5.B=51 and
  R1.B=R2.B and
  R2.B=R3.B and
  R3.B=R4.B and
  R4.B=R5.B;
```



selected	order	from
1	0	4 SCAN TABLE R5
2	1	3 SEARCH TABLE R4 USING AUTOMATIC COVERING INDEX [B=?]
3	2	2 SEARCH TABLE R3 USING AUTOMATIC COVERING INDEX [B=?]
4	3	1 SEARCH TABLE R2 USING AUTOMATIC COVERING INDEX [B=?]
5	4	0 SEARCH TABLE R1 USING AUTOMATIC COVERING INDEX [B=?]

Each table returned the same exact table, which was this.

Query Execution: These are the rows that were returned from each of the queries above.

–See Attached files for each plan. Plan(1–5).tsv (They’re huge and don’t make sense to attach to this document)

```
sql> select *
from R1, R2, R3, R4, R5
where
    R1.B=R2.B and R2.B=R3.B and R3.B=R4.B and R4.B=R5.B and R5.B=51
[2018-04-04 23:34:06] 448 rows retrieved starting from 1 in 68ms (execution: 6ms, fetching: 60ms)
sql> select *
from R1, R2, R3, R4, R5
where R5.B=51 and
    R1.B=R2.B and R2.B=R3.B and
    R3.B=R4.B and R4.B=R5.B
[2018-04-04 23:34:07] 448 rows retrieved starting from 1 in 326ms (execution: 7ms, fetching: 319ms)
sql> select *
from R5, R1, R2, R3, R4
where
    R5.B=51 and R1.B=R2.B and R2.B=R3.B and
    R3.B=R4.B and R4.B=R5.B
[2018-04-04 23:34:07] 448 rows retrieved starting from 1 in 328ms (execution: 7ms, fetching: 321ms)
sql> select *
from R1,R2, R4,R3, R5
where
    R1.B=R2.B and R2.B=R3.B and
    R3.B=R4.B and R4.B=R5.B and not ( R5.B < 51 or R5.B > 51)
[2018-04-04 23:34:07] 448 rows retrieved starting from 1 in 367ms (execution: 23ms, fetching: 344ms)
sql> select /*+NO_MERGE*/ *
from R1, R3, R2, R4, R5
where
    R5.B=51 and
    R1.B=R2.B and
    R2.B=R3.B and
    R3.B=R4.B and
    R4.B=R5.B
[2018-04-04 23:34:08] 448 rows retrieved starting from 1 in 336ms (execution: 6ms, fetching: 330ms)
```

Conclusion:

Because of my problems trying to execute the EXPLAIN PLAN on the command line, I feel like I missed out on some important data. I still feel like I understand how the explain works and how the joins are being computed, but it would be nice to have the data available. One observation that I did make is that in some of the queries, the columns and rows appear in different orders. For example Plan1 and Plan3 have different columns for column 1 and column 2. That leads me to assume that they were executed in a different manner due based on the SQL optimizer. I was a bit surprised by how quickly the queries were computed, as it took only milliseconds to do the join on all five relations. From the data above, it appears that Plan1 is the best as it only took 68 milliseconds to run the entire query.

NOTEThe first time I created the data, the number 51 didn’t appear in all of the tables, so I added it myself to random rows in each table.***END NOTE***